



COMP781

Advanced Algorithms

S1 Day 2015

Dept of Computing

Contents

| | |
|--|---|
| <u>General Information</u> | 2 |
| <u>Learning Outcomes</u> | 2 |
| <u>Assessment Tasks</u> | 3 |
| <u>Delivery and Resources</u> | 4 |
| <u>Unit Schedule</u> | 5 |
| <u>Policies and Procedures</u> | 5 |
| <u>Graduate Capabilities</u> | 6 |
| <u>Assessment Standards</u> | 8 |

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff

Convenor

Annabelle McIver

annabelle.mciver@mq.edu.au

E6A 379

Lecturer

Franck Cassez

franck.cassez@mq.edu.au

E6A 313

Credit points

4

Prerequisites

Admission to MRes

Corequisites

Co-badged status

Unit description

Algorithms are the essence of computer science. In this unit we build on the undergraduate understanding of algorithms and look at interesting and useful algorithms, both fundamental and cutting edge. The particular material covered will depend on the cohort but may include topics such as approximation algorithms, exponential-time exact and parameterized algorithms, linear and constraint programming and fundamental graph algorithms such as max-flow algorithms, matching algorithms and so on. The unit will also employ appropriate tools from complexity theory to analyse the performance of the algorithms studied.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at <https://www.mq.edu.au/study/calendar-of-dates>

Learning Outcomes

On successful completion of this unit, you will be able to:

Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.

Understand and formally analyse algorithms.

Implement key algorithms.

Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.

Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment Tasks

| Name | Weighting | Due |
|---------------------|-----------|-------------------|
| <u>Presentation</u> | 10% | Week 13 |
| <u>Weekly Tests</u> | 36% | Week 1 to Week 12 |
| <u>Projects</u> | 54% | Weeks 3--12 |

Presentation

Due: **Week 13**

Weighting: **10%**

An oral presentation supported by appropriate presentation materials will be conducted in week 13.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Weekly Tests

Due: **Week 1 to Week 12**

Weighting: **36%**

Each week students will be asked to complete some exercises in class to test understanding of the material.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse algorithms.

Projects

Due: **Weeks 3--12**

Weighting: **54%**

Students will be asked to complete 4 projects. These will consist of a combination of programming, program analysis and report writing.

On successful completion you will be able to:

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Understand and formally analyse algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Delivery and Resources

Classes

Each week has three hours of face-to-face class. These classes will be a mixture lecture material, discussion and in class tests.

Recommended Reading and References

There is no set text for the course, but the following *far from exhaustive* list of texts may be useful for reference, study and further reading:

- Skiena, Algorithm Design Manual, Springer.
- Cormen, Leiserson, Rivest and Stein. Introductions to algorithms, Prentice Hall.
- Papadimitriou, Computational Complexity, Addison Wesley.
- Sipser, Introduction to the Theory of Computation, Thomson.

Unit Webpage, Materials and Technologies Used

The materials for the unit including notes, discussion fora, electronic submission links etc. will be through the iLearn system.

The programming projects can be done in any programming language subject to prior approval of the course convener. Languages can include Java and Python.

Unit Schedule

Algorithms are the essence of computer science. In this unit we build on the undergraduate understanding of algorithms and look at interesting and useful algorithms, both fundamental and cutting edge.

Throughout the unit there is an emphasis on developing not just a knowledge of algorithms, but of general algorithmic techniques alongside a deeper understanding of why algorithms work at a formal level. This helps facilitate the student's ability to construct a 'toolkit' of algorithmic techniques that can be adapted and employed to solve computational problems in their ongoing studies and work.

Part of this understanding will be the ongoing development of an intuition and formal approaches for comparing and evaluating algorithms.

Students entering this unit should have programming experience commensurate with the completion of an undergraduate computing degree, a working knowledge of basic and intermediate algorithms and experience with discrete mathematics. Further algorithmic and mathematical study will be of benefit, but is not necessary. Directly relevant undergraduate units offered at Macquarie include COMP225 - Algorithms and Data Structures, DMTH237 - Discrete Mathematics II and their prerequisites. Completion of COMP333 - Algorithm Theory and Design would be helpful.

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central](#). Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy <http://mq.edu.au/policy/docs/assessment/policy.html>

Grading Policy <http://mq.edu.au/policy/docs/grading/policy.html>

Grade Appeal Policy <http://mq.edu.au/policy/docs/gradeappeal/policy.html>

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Disruption to Studies Policy http://www.mq.edu.au/policy/docs/disruption_studies/policy.html *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

In addition, a number of other policies can be found in the [Learning and Teaching Category](#) of Policy Central.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/support/student_conduct/

Results

Results shown in *iLearn*, or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in [eStudent](#). For more information visit [ask.mq.edu.au](#).

Student Support

Macquarie University provides a range of support services for students. For details, visit <http://students.mq.edu.au/support/>

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- [Workshops](#)
- [StudyWise](#)
- [Academic Integrity Module for Students](#)
- [Ask a Learning Adviser](#)

Student Services and Support

Students with a disability are encouraged to contact the [Disability Service](#) who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

IT Help

For help with University computer systems and technology, visit <http://informatics.mq.edu.au/help/>.

When using the University's IT, you must adhere to the [Acceptable Use Policy](#). The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

PG - Capable of Professional and Personal Judgment and Initiative

Our postgraduates will demonstrate a high standard of discernment and common sense in their professional and personal judgment. They will have the ability to make informed choices and decisions that reflect both the nature of their professional work and their personal perspectives.

This graduate capability is supported by:

Assessment task

- Presentation

PG - Discipline Knowledge and Skills

Our postgraduates will be able to demonstrate a significantly enhanced depth and breadth of knowledge, scholarly understanding, and specific subject content knowledge in their chosen fields.

This graduate capability is supported by:

Learning outcomes

- Understand and formally analyse algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment tasks

- Weekly Tests
- Projects

PG - Critical, Analytical and Integrative Thinking

Our postgraduates will be capable of utilising and reflecting on prior knowledge and experience, of applying higher level critical thinking skills, and of integrating and synthesising learning and knowledge from a range of sources and environments. A characteristic of this form of thinking is the generation of new, professionally oriented knowledge through personal or group-based critique of practice and theory.

This graduate capability is supported by:

Learning outcomes

- Understand and formally analyse algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment tasks

- Weekly Tests

- Projects

PG - Research and Problem Solving Capability

Our postgraduates will be capable of systematic enquiry; able to use research skills to create new knowledge that can be applied to real world issues, or contribute to a field of study or practice to enhance society. They will be capable of creative questioning, problem finding and problem solving.

This graduate capability is supported by:

Learning outcomes

- Understand and formally analyse algorithms.
- Implement key algorithms.
- Develop algorithmic solutions for computational problems by constructing new algorithms and combining existing algorithms.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment tasks

- Weekly Tests
- Projects

PG - Effective Communication

Our postgraduates will be able to communicate effectively and convey their views to different social, cultural, and professional audiences. They will be able to use a variety of technologically supported media to communicate with empathy using a range of written, spoken or visual formats.

This graduate capability is supported by:

Learning outcomes

- Explain key ideas in the field of algorithmics and the workings of key algorithms, and compare and evaluate algorithmic solutions for computational problems.
- Investigate topics in advanced algorithms and synthesise the output for presentation in oral and written form.

Assessment tasks

- Presentation
- Projects

Assessment Standards

COMP781 will be assessed and graded according to the University assessment and grading

policies.

Submission Deadlines

Assessment deadlines are strict, unless an application for special consideration is received (preferably in advance) accompanied by appropriate documentary evidence. Late submissions will be penalised at the rate of 20% of the full marks for the assessment per day or part thereof late.

Standards

The following general standards of achievement will be used to assess each of the assessment tasks with respect to the letter grades.

Pass: Has a basic understanding of the algorithms and concepts as discussed in class. Can describe and reproduce definitions and fundamental algorithms. Can perform a basic research investigation in the area and present the results of that research in rudimentary written and oral forms.

Credit: As for Pass plus: Is able to apply the algorithmic techniques we have discussed to derive solutions to computational problems. Can develop, generalise and apply the concepts discussed in class to address basic theoretical and practical questions, and can effectively communicate these insights. Shows more than basic insights into the results of a research investigation and is able to communicate those insights.

Distinction/High Distinction: As for Credit plus: Is able to generalise and synthesise knowledge to address more complex topics beyond the material discussed in class. Can critically evaluate the limits of the techniques and algorithms discussed.

Assessment Process

These assessment standards will be used to give a numeric mark out of 100 to each assessment submission during marking. The mark will correspond to a letter grade for that task according to the University guidelines. The final raw mark for the unit will be calculated by combining the marks for all assessment tasks according to the percentage weightings shown in the assessment summary.

We will look at your overall performance on all assessments when determining your final grade. A total raw mark of at least 50% and a mark of at least 50% on each of the assessment categories will be sufficient to pass the unit. Students who do not meet this cut-off will be examined on a case-by-case basis.

On occasion your raw mark for the unit may not be the same as the Standardised Numeric Grade (SNG) which you receive as the result. Under the Senate guidelines, raw marks may be scaled to ensure that there is a degree of comparability across the university, so that units with the same past performances of their students should achieve similar results.